

# A Note on Nonuniform versus Uniform $\text{ACC}^k$ Circuits for $\text{NE}^*$

Lane A. Hemaspaandra<sup>†</sup>  
Department of Computer Science  
University of Rochester  
Rochester, NY 14627, USA

December 2, 2010; revised December 3, 2010

## Abstract

We note that for each  $k \in \{0, 1, 2, \dots\}$  the following holds:  $\text{NE}$  has (nonuniform)  $\text{ACC}^k$  circuits if and only if  $\text{NE}$  has  $\text{P}^{\text{NE}}$ -uniform  $\text{ACC}^k$  circuits. And we mention how to get analogous results for other circuit and complexity classes.

## 1 Introduction and Result

Ryan Williams recently announced the breakthrough advance that some  $\text{NE}$  sets lack  $\text{ACC}^0$  circuits [Wil10]. His result is for the extremely strong case of defeating even nonuniform  $\text{ACC}^0$  circuits.

Is there some on-the-surface-weaker claim—about defeating *uniform*  $\text{ACC}^0$  circuits—that is equivalent to this? This brief note looks at that question, for the case of each  $\text{ACC}^k$ . What we observe is the following.

**Theorem 1.1** *For each  $k \in \{0, 1, 2, \dots\}$  the following holds:  $\text{NE}$  has (nonuniform)  $\text{ACC}^k$  circuits if and only if  $\text{NE}$  has  $\text{P}^{\text{NE}}$ -uniform  $\text{ACC}^k$  circuits.*

The immediate natural question to ask is: Why should one care about this? After all, for  $k = 0$  Williams’s result already handles the most challenging case, nonuniform  $\text{ACC}^0$ , and the above result simply let one conclude, from his result, a far weaker result. However, there are two related reasons why one should care about the above theorem. First, regarding  $k = 0$ , the goal of the above result isn’t to extend Williams’s result, but rather is to understand it better, and in particular, to understand what seemingly weaker uniform result—which,

---

\*Also appears as URCS-TR-2010-964

<sup>†</sup>URL: [www.cs.rochester.edu/u/lane](http://www.cs.rochester.edu/u/lane). Supported in part by grant NSF-CCF-0915792 and a Friedrich Wilhelm Bessel Research Award.

we should stress, no one ever obtained—would have implied Williams’s nonuniform result. That is, instead of pole vaulting over a 10-meter-high bar and shattering the world record, Williams could have indirectly achieved the same strength-of-result by pole vaulting over a bar that was merely 9.99 meters high. Second and more important, for  $k > 0$ , the above result potentially puts in place a very slightly lower bar for whoever tries to show that, for example, nonuniform  $\text{ACC}^1$  circuits cannot handle all of NE. Instead of trying to defeat nonuniform circuits, that researcher need only (although that is a huge “only”) defeat  $\text{P}^{\text{NE}}$ -uniform circuits. And although the above theorem says that that is logically the same as defeating nonuniform  $\text{ACC}^1$  circuits, it is quite possible that one can more easily (although that is unlikely to be an easy “easily”) argue regarding the limitations of circuits of limited-complexity uniformity than one can about nonuniform circuits. Although we won’t repeat here the history and background that are well-covered in the paper of Williams, it is worth mentioning that some of the results that proceeded Williams’s work centrally used the uniformity of the classes being defeated (see the work of Allender and Gore [AG94, All99]).

The proof of the theorem is quite brief. One just takes the brute-force bound (please see footnote 1 for history and for credit/relation to Hopcroft) one gets from guessing and checking circuits,<sup>1</sup> sees that one gets a bound at the  $\Delta_4$  function-level of the so-called strong

---

<sup>1</sup>Guessing and checking is an often helpful approach in complexity theory. It is employed in Williams’s proof [Wil10], and it has a long history, e.g., one can find it in Hopcroft’s alternate proof of the Karp–Lipton Theorem ([Hop81], and as that is a conference-length-only paper and may be a bit hard to find, please note that a more recent writeup of that proof that very explicitly and in detail follows the guess-and-check approach of Hopcroft can be found as [HO02, Proof of Theorem 1.16]) and in the work of Balcázar, Book, Long, Schöning, and Selman [BBS86, LS86] showing that the polynomial hierarchy collapses if and only if the polynomial hierarchy collapses relative to some sparse set if and only if the polynomial hierarchy collapses relative to every sparse set.

In fact, it is important as credit-where-credit-is-due to mention that in a very real sense the Hopcroft insight mentioned above *is* (or is very close to) the same “brute-force” approach we’re focusing on in this note (although the point of this note is mostly to note that for the case of NE an unexpected simplification occurs, and also to make explicit how the brute-force approach plays out in a circuit-uniformity setting; however, in fact, in some ways, the Hopcroft work is better than brute force—see below—as it uses an additional wonderful trick specific to its own setting).

The following rather technical expansion on the comment just made is addressed only to those who are familiar with the Hopcroft approach.

What we mean when we say that the Hopcroft “guess the sparse set (or circuit, as  $\text{P}^{\text{SPARSE}} = \text{P}/\text{poly}$  and so guessing sparse sets and guessing a small general circuit are almost the same, in this context) and check” proof of the Karp–Lipton Theorem really is doing the same thing, or very, very closely to the same thing, as the brute-force approach mentioned here is that the only real difference (when one looks beyond the surface and thanks about the flavor of what is going on) is that in the circuit uniformity case one is *producing* a circuit, and thus we in our oracle stack of classes have an FP on bottom, but in contrast Hopcroft’s proof can have lots of different paths that guess good sparse sets (or circuits) and that is no problem there as they all will do the right thing and there will be at least one such path; and a second difference is that Hopcroft actually uses *less* of a stack of quantifier access (even aside from the extra FP on bottom for the reason just mentioned), because he isn’t merely brute-forcing things, but is using the utterly lovely trick of the 2-disjunctive-self-reducibility of SAT, which lets him with a single “forall”-type oracle call check the consistency of each internal node of the self-reducibility tree and also check the leaves and by doing so already know if the given path has a good sparse set (or circuit), and if so then that path then uses that setting’s own  $\text{NP}^{\text{NP}}$  to handle the first two levels of the target  $\text{NP}^{\text{NP}^{\text{NP}}}$  set while passing up (along with

exponential hierarchy, and then invokes a result from the 1980s that shows that the strong exponential hierarchy (whose levels are  $E$ ,  $NE$ ,  $P^{NE}$ ,  $NP^{NE}$ ,  $P^{NP^{NE}}$ ,  $NP^{NP^{NE}}$ ,  $\dots$ )<sup>2</sup> collapses to its  $\Delta_2$  level [Hem89]. As to that brute-force bound, for any typical, reasonable class  $\mathcal{D}$  of polynomial-size circuits, and any reasonable complexity class  $\mathcal{C}$ , the brute-force bound one gets is that if  $\mathcal{C}$  has nonuniform circuits in  $\mathcal{D}$ , then  $\mathcal{C}$  has  $P^{NP^{coNP^{\mathcal{C}}}}$ -uniform circuits in  $\mathcal{D}$  (equivalently, of course, has  $P^{NP^{NP^{\mathcal{C}}}}$ -uniform circuits in  $\mathcal{D}$ ).<sup>3</sup> This is true in the obvious way. The uniformity class is asserting that on input  $1^j$  the  $FP^{NP^{coNP^{\mathcal{C}}}}$  function outputs the circuit (description) for the circuit that handles length- $j$  inputs. Briefly put, this is done as follows. First, let  $A$  be an arbitrary  $NE$  set that we assume has (nonuniform) circuits in  $\mathcal{D}$ . Let  $q$  be the polynomial bound on the size of those circuits (or to be more precise, their descriptions), and if they have a depth bound, let it be realized by the constant-or-function  $d$  (e.g., for  $ACC^0$ ,  $d(j)$  would be any particular constant by which the hypothetical  $ACC^0$  circuits for  $A$  were depth-bounded). The  $FP$  part conducts a prefix search to find the first (say, in dictionary ordering)  $\mathcal{D}$ -type (i.e., within the allowed depth  $d(j)$ ,  $q(j)$ -size-bounded, and with the right types of gates and fan-in, etc.) circuit that is a correct circuit for  $A$  for all length- $j$  inputs. The  $NP^{coNP^{\mathcal{C}}}$  part supports this in two ways (which way it is being used on a particular call will be specified by an extra bit, not explicitly mentioned below, of the call's argument string). It can answer the  $coNP^{\mathcal{C}}$  question: Here is your input, which is  $1^j$

---

other things) the sparse set (or circuit) so that the upper of the two  $NP$ 's can use the sparse set (or circuit) to handle the top-most  $NP$  of the height-3 target set, and that works perfectly in his setting. That latter difference is not a generally available trick, but it works like magic in his  $\mathcal{C} = NP$  case, and actually, in his setting that magic seems needed to get the collapse from the proof approach (although historically there already was the quite different Karp-Lipton proof).

<sup>2</sup>As a brief bit of context, taken from [Hem89], we mention that by padding,  $P^{NE} = P^{NEXP}$  (recall  $NE$  is  $NTIME[2^{O(n)}]$  and  $NEXP$  is  $NTIME[2^{O(n^{O(1)})}]$ ), and that  $P^{NE} \subseteq EXP^{NP}$  but it is not at all clear that the reverse containment holds, although the reverse containment is known to hold if  $EXP$  only accesses its oracle nonadaptively.

<sup>3</sup>Recall that for example, the literature term  $P$ -uniform actually truly is speaking not of  $P$  but is speaking of  $FP$ , the polynomial-time computable functions, since we are speaking of which class is used to on input  $1^j$  output an appropriate circuit to handle all inputs of length  $j$ , and that is a function issue. But following that same convention, we speak of  $P^{NP^{coNP^{\mathcal{C}}}}$ -uniform, meaning that the function generating the circuits is actually in the class  $FP^{NP^{coNP^{\mathcal{C}}}}$ -uniform (and similarly, by  $P^{NE}$ -uniform, we mean that the circuit generator is in  $FP^{NE}$ ). We mention in passing that if one wants to make a slightly stronger claim, one can assert under the same assumption the existence of—here we really will name a function class,  $OptP$  [Kre88], directly— $OptP^{NP^{\mathcal{C}}}$ -uniform circuits in  $\mathcal{D}$ . That slight improvement would not help us regarding Theorem 1.1 since an easy consequence of the collapse of the strong exponential hierarchy is that each  $OptP^{NP^{NE}}$  function in fact is in  $FP^{NE}$ .

Eric Allender (personal communication, November 30, 2010) interestingly pointed that one can go in the opposite direction, namely, that the in general *more* powerful class  $FSPACE_{poly}^{\mathcal{C}}$  (where the “poly” means that the  $FSPACE$  machine asks only polynomially long queries to its oracle) can by brute-force cycle through circuits and inputs and do the appropriate checking. Although that class in general may be larger, and so one would not want to use this in such cases (in fact, similarly, the  $OptP^{\mathcal{C}}$  path noted above may well in some settings yield better claims than even using  $FP^{NP^{coNP^{\mathcal{C}}}}$ ), Eric notes that for the case  $\mathcal{C} = NE$  it is not larger because the collapse of the strong exponential hierarchy has been extended (see [SW88,Hem94]) to yield  $PSPACE_{poly}^{NE} = P^{NE}$  (and even  $NEXP_{poly}^{NE} = P^{NE}$ ).

and a circuit (description), and does that described circuit correctly match  $A$  in terms of acceptance/rejection on all inputs of length  $j$ ? And it can also answer the  $\text{NP}^{\text{coNP}^c}$  question: Here is  $1^j$  and a string  $\alpha$  as your input, and please let me know whether there exists a string  $\beta$  such that the combined length of  $\alpha$  and  $\beta$  is at most  $q(j)$  and the concatenation of  $\alpha$  and  $\beta$  is a circuit (description) of  $\mathcal{D}$ -type (i.e., is within the allowed depth  $d(j)$ ,  $q(j)$ -size-bounded, and uses the allowed types of gates and fan-in, etc.) that correctly matches  $A$  in terms of acceptance/rejection on all inputs of length  $j$ ? (Regarding this latter use, the NP part is guessing the completion of the circuit, the coNP part is ranging over all length- $j$  strings, and for each such string  $y$  is seeing what the circuit does (we're assuming our circuits are such that with their description in hand we can in polynomial time evaluate the circuit's action on a given string) on  $y$  and then through one query to  $\mathcal{C}$  is finding whether  $y \in A$ , and then on that path of the coNP machine accepts if the two actions agree—our coNP machine model is that the machine by definition accepts exactly if all of its paths accept.)

For the particular case of  $\text{ACC}^k$  and NE, this gives that if NE has (nonuniform)  $\text{ACC}^k$  circuits then there is a  $\text{FP}^{\text{NP}^{\text{coNP}^{\text{NE}}}}$  function that on input  $1^j$  generates a good circuit for length- $j$  inputs. However, trivially,  $\text{FP}^{\text{NP}^{\text{coNP}^{\text{NE}}}} = \text{FP}^{\text{NP}^{\text{NP}^{\text{NE}}}}$ , and since the collapse of the strong exponential hierarchy [Hem89] yields  $\text{NP}^{\text{NP}^{\text{NE}}} = \text{P}^{\text{NE}}$ , we have  $\text{FP}^{\text{NP}^{\text{coNP}^{\text{NE}}}} = \text{FP}^{\text{P}^{\text{NE}}} = \text{FP}^{\text{NE}}$ , i.e., we have  $\text{P}^{\text{NE}}$ -uniformity.

**Acknowledgments** All errors and blindnesses are the sole responsibility of the author, who is not a specialist in circuit theory. The author thanks Eric Allender for the comments mentioned in footnote 3 and for helpful discussions, Daniel Štefankovič for a helpful conversation about Williams's result, and Ryan Williams for pointing him to the reference [All99].

## References

- [AG94] E. Allender and V. Gore. A uniform circuit lower bound for the permanent. *SIAM Journal on Computing*, 23(5):1026–1049, 1994.
- [All99] E. Allender. The permanent requires large uniform threshold circuits. *Chicago Journal of Theoretical Computer Science*, Volume 1999 (Article 7), 1999.
- [BBS86] J. Balcázar, R. Book, and U. Schöning. The polynomial-time hierarchy and sparse oracles. *Journal of the ACM*, 33(3):603–617, 1986.
- [Hem89] L. Hemachandra. The strong exponential hierarchy collapses. *Journal of Computer and System Sciences*, 39(3):299–322, 1989.
- [Hem94] E. Hemaspaandra. Census techniques collapse space classes. *Information Processing Letters*, 51(2):79–84, 1994.
- [HO02] L. Hemaspaandra and M. Ogihara. *The Complexity Theory Companion*. Springer-Verlag, 2002.

- [Hop81] J. Hopcroft. Recent directions in algorithmic research. In *Proceedings 5th GI Conference on Theoretical Computer Science*, pages 123–134. Springer-Verlag *Lecture Notes in Computer Science #104*, March 1981.
- [Kre88] M. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36(3):490–509, 1988.
- [LS86] T. Long and A. Selman. Relativizing complexity classes with sparse oracles. *Journal of the ACM*, 33(3):618–627, 1986.
- [SW88] U. Schöning and K. Wagner. Collapsing oracle hierarchies, census functions, and logarithmically many queries. In *Proceedings of the 5th Annual Symposium on Theoretical Aspects of Computer Science*. Springer-Verlag *Lecture Notes in Computer Science*, February 1988.
- [Wil10] R. Williams. Non-uniform ACC circuit lower bounds. Manuscript, available online at [www.cs.cmu.edu/~ryanw/acc-lbs.pdf](http://www.cs.cmu.edu/~ryanw/acc-lbs.pdf), November 23, 2010.